Project 3: Classification

Vandana Prasad Gaallee/50289877

November 2018

1 Introduction

This project is to implement machine learning methods for the task of classification of 28*28 grayscale handwritten digit image and identify it as a digit among 0, 1, 2, ... , 9. This problem is implemented using Logistic regression, Random Forest, Neural Networks and Support Vector Machine algorithms.

2 Data Extraction

The classification models are trained using MNIST dataset and tested using both MNIST and USPS dataset.

MNIST Dataset : The MNIST dataset has 70,000 input sample with 784 feature vectors. 80 percent of input samples is used for training and 20 percent is used for testing and validation.

USPS Dataset : The USPS dataset has 20,000 input sample with 784 feature vectors. This data is tested against MNIST trained model and performance is recorded.

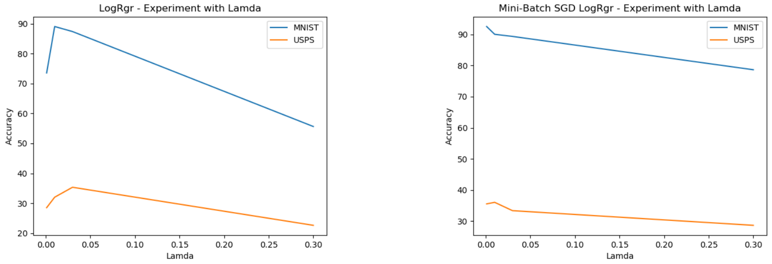
3 Logistic Regression

Logistic Regression is a machine learning classification algorithm that is used to predict the ouput given a input. It is implemented using Gradient Descent and the model is trained using MNIST dataset. It is given by formula

$$y(x, w) = Softmax(w^T X) \tag{1}$$

3.1 Results on changing HyperParameters

1. **Case 1 : Experimenting with HyperParameter Lamda** Below is the graph representing the change in accuracy with respect to lamda for USPS and MNIST dataset. The accuracy is compared with lambda values for both gradient descent and Mini-batch SGD implementations



(a) label 1

(b) label 2

Figure 1: Accuracy wrt Lamda

2. **Case 2 : Experimenting with HyperParameter Learning rate** Below graph represents the change in accuracy with respect to learning rate.

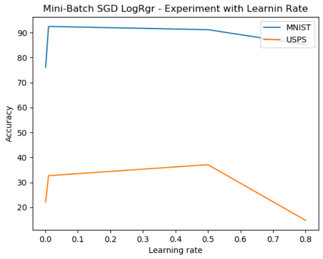
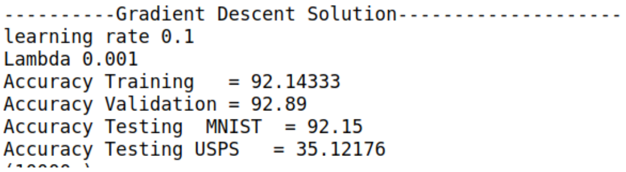


Figure 2: Accuracy with respect to Learning rate

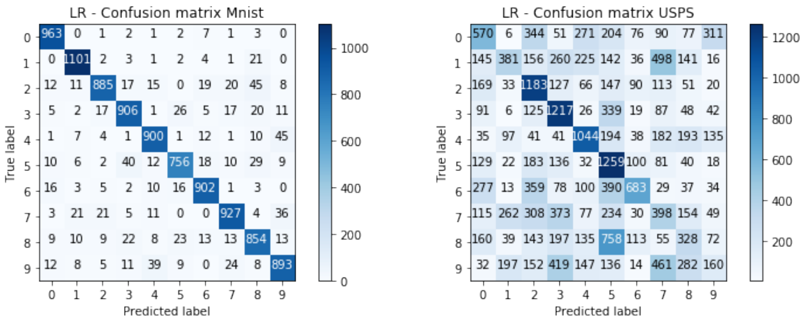
3.2 Results Obtained

Accuracy obtained for Logistic regression using MNIST test dataset is significantly higher than USPS dataset. Mini-batch SGD performs better than gradient descent implementation.



```
----------Gradient Descent Solution--------------------
learning rate 0.1
Lambda 0.001
Accuracy Training   = 92.14333
Accuracy Validation = 92.89
Accuracy Testing  MNIST  = 92.15
Accuracy Testing USPS   = 35.12176
```

Figure 3: Accuracy - Logistic Regression



(a) MNIST

(b) USPS

Figure 4: Confusion Matrix

4 Neural Networks

The given classification problem is implemented using neural networks with multiple layers and performance in each case is recorded.

4.1 Results on changing HyperParameters

1. **Case 1 : Experimenting with Layers** The performance of model improved with initial increase in layers, However it degraded with more complex network. Below is the graph depicting the variations in accuracy with respect the change in layers.

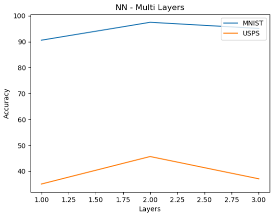


Figure 5: Accuracy with change in Layers

2. **Case 2 : Experimenting with HyperParameter Dropout** Below graph represents the change in accuracy with respect to dropout rate.

4.2 Results Obtained

NN solution performs better than all other implementation and gives an accuracy of 97.5% for MNIST dataset and 45% for USPS test dataset.

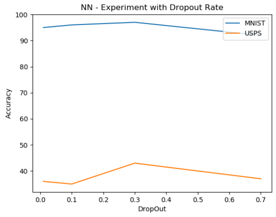
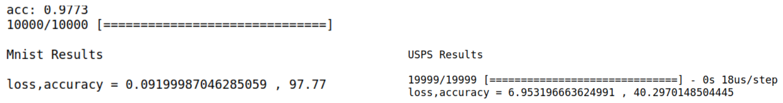


Figure 6: Accuracy with respect to Dropout rate



```
acc: 0.9773
10000/10000 [==============================]

Mnist Results

loss,accuracy = 0.09199987046285059 , 97.77
```

(a) MNIST

```
USPS Results

19999/19999 [==============================] - 0s 18us/step
loss,accuracy = 6.953196663624991 , 40.2970148504445
```

(b) USPS

Figure 7: Accuracy

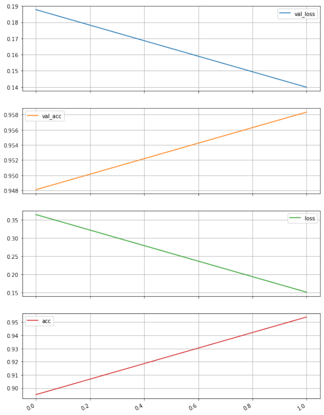
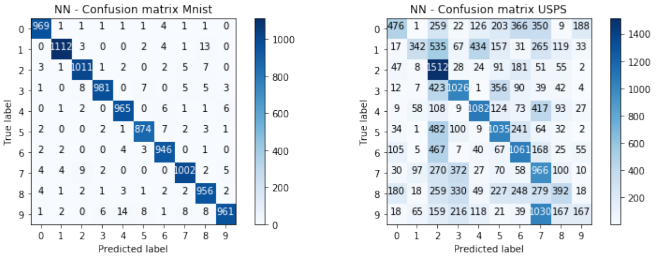


Figure 8: Accuracy and Loss



NN - Confusion matrix Mnist

	0	1	2	3	4	5	6	7	8	9
0	969	1	1	1	1	1	4	1	1	0
1	0	1112	3	0	0	2	4	1	13	0
2	3	1	1011	1	2	0	2	5	7	0
3	1	0	8	981	0	7	0	5	5	3
4	0	1	2	0	965	0	6	1	1	6
5	2	0	0	2	1	874	7	2	3	1
6	2	2	0	0	4	3	946	0	1	0
7	4	4	9	2	0	0	0	1002	2	5
8	4	1	2	1	3	1	2	2	956	2
9	1	2	0	6	14	8	1	8	8	961

(a) MNIST

NN - Confusion matrix USPS

	0	1	2	3	4	5	6	7	8	9
0	476	1	259	22	126	203	366	350	9	188
1	17	342	535	67	434	157	31	265	119	33
2	47	8	1512	28	24	91	181	51	55	2
3	12	7	423	1026	1	356	90	39	42	4
4	9	58	108	9	1082	124	73	417	93	27
5	34	1	482	100	9	1035	241	64	32	2
6	105	5	467	7	40	67	1061	168	25	55
7	30	97	270	372	27	70	58	966	100	10
8	180	18	259	330	49	227	248	279	392	18
9	18	65	159	216	118	21	39	1030	167	167

(b) USPS

Figure 9: Confusion Matrix

5 Random Forest

The given classification problem is implemented using Random Forest classification problem using sklearn RandomForestClassifier package. A random forest is an estimator which fits multiple decision tree classifiers over sub-samples of dataset.

5.1 Experimenting with HyperParameter Estimator

Estimators is the measure of number of trees in the forest and the accuracy of the model increased steadily with increase in number of estimators. Below is the graph depicting the variations in accuracy with respect the change in estimators.

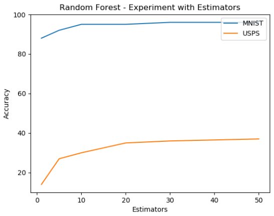
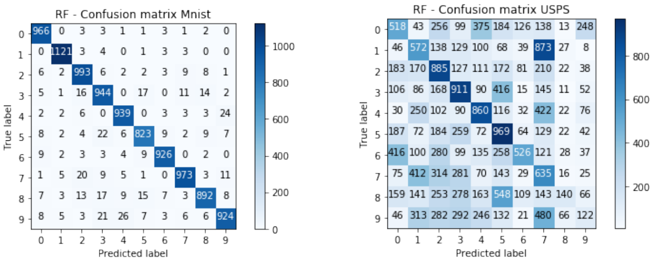


Figure 10: Accuracy with change in Estimators

5.2 Confusion Matrix


RF - Confusion matrix Mnist

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	966	0	3	3	1	1	3	1	2	0
1	0	1121	3	4	0	1	3	3	0	0
2	6	2	993	6	2	2	3	9	8	1
3	5	1	16	944	0	17	0	11	14	2
4	2	2	6	0	939	0	3	3	3	24
5	8	2	4	22	6	823	9	2	9	7
6	9	2	3	3	4	9	926	0	2	0
7	1	5	20	9	5	1	0	973	3	11
8	7	3	13	17	9	15	7	3	892	8
9	8	5	3	21	26	7	3	6	6	924

(a) MNIST

RF - Confusion matrix USPS

True label \ Predicted label	0	1	2	3	4	5	6	7	8	9
0	518	43	256	99	375	184	126	138	13	248
1	46	572	138	129	100	68	39	873	27	8
2	183	170	885	127	111	172	81	210	22	38
3	106	86	168	911	90	416	15	145	11	52
4	30	250	102	90	860	116	32	422	22	76
5	187	72	184	259	72	969	64	129	22	42
6	416	100	280	99	135	258	526	121	28	37
7	75	412	314	281	70	143	29	635	16	25
8	159	141	253	278	163	548	109	143	140	66
9	46	313	282	292	246	132	21	480	66	122

(b) USPS

Figure 11: Confusion Matrix

5.3 Results Obtained

RF solution performs better than logistic regression. Below is a table showing the accuracy obtained for MNIST and USPS test dataset

Dataset	Accuracy
MNIST	96.2
USPS	36.4

6 Support Vector Machine

The given classification problem is implemented sklearn.svm package.

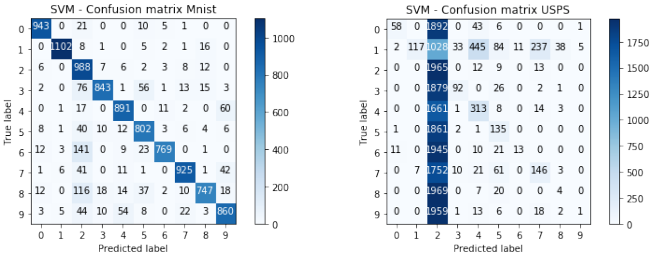
6.1 Experimenting with different settings

1. **case 1:** Using linear kernel (all other parameters are kept default).
   Accuracy Obtained : 95.43

2. **case 2:** Using radial basis function with value of gamma setting to 1 (all other parameters are kept default).
   Accuracy obtained : 17.56

3. **case 3:** Using radial basis function with value of gamma setting to default (all other parameters are kept default).
   Accuracy obtained : 94.34

6.2 Confusion Matrix



(a) MNIST

(b) USPS

Figure 12: Confusion Matrix

6.3 Results Obtained

Support Vector Machine solution performs better than logistic regression and performs poor over USPS datset. Below is a table showing the accuracy obtained for MNIST and USPS test dataset.

Dataset	Accuracy
MNIST	96.65
USPS	20.98

7 Inferences

7.1 Compliance with “No Free Lunch” theorem

The “No Free Lunch” theorem states that there is no one model that works best for every problem. This is very evident from the results of USPS test dataset with accuracy of no more than 45% over the model trained with MNIST dataset. Below is graph comparing the accuracy obtained using USPS and MNIST dataset using different classifiers.

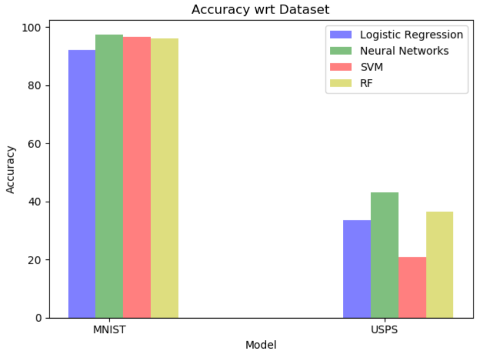


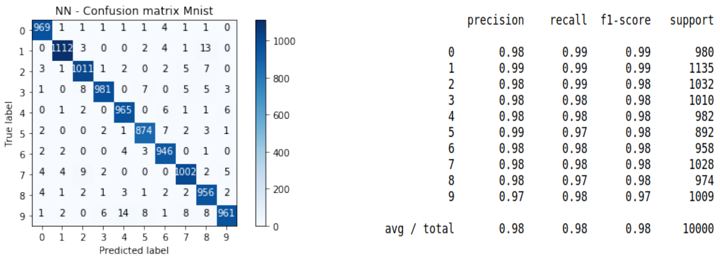
Figure 13: Accuracy and Loss

7.2 Analysis of Confusion Matrix

Confusion Matrix is generated for each of the classifiers using sklearn.metrics confusion_matrix package. It is used to evaluate the quality of the output of a classifier.

1. The diagonal elements represent the number of points for which the predicted label is equal to the true label.
2. The off-diagonal elements are those that are mislabeled by the classifier.
3. The higher the diagonal values of the confusion matrix the better, indicating many correct predictions.

This is evident from the below comfusion matrix and classificatiion report of Neural networks. The diagnol elements are high which represent the true positives. The classification report represents the precision and recall scores. Precision is the number of correct predictions of class over total number of predictions of the class. Recall is the number of predictions of class over total number of expected predictions.



	0	1	2	3	4	5	6	7	8	9
0	969	1	1	1	1	1	4	1	1	0
1	0	1112	3	0	0	2	4	1	13	0
2	3	1	1011	1	2	0	2	5	7	0
3	1	0	8	981	0	7	0	5	5	3
4	0	1	2	0	965	0	6	1	1	6
5	2	0	0	2	1	874	7	2	3	1
6	2	2	0	0	4	3	946	0	1	0
7	4	4	9	2	0	0	0	1002	2	5
8	4	1	2	1	3	1	2	2	956	2
9	1	2	0	6	14	8	1	8	8	961

(a) Confusion Matrix

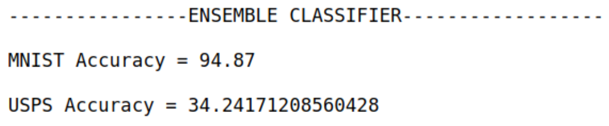
```
             precision    recall  f1-score   support

          0       0.98      0.99      0.99       980
          1       0.99      0.99      0.99      1135
          2       0.98      0.99      0.98      1032
          3       0.98      0.98      0.98      1010
          4       0.98      0.98      0.98       982
          5       0.99      0.97      0.98       892
          6       0.98      0.98      0.98       958
          7       0.98      0.98      0.98      1028
          8       0.98      0.97      0.98       974
          9       0.97      0.98      0.97      1009

avg / total       0.98      0.98      0.98     10000
```

(b) Classification Report

Figure 14: Neural Networks

7.3 Ensemble Classifier

Ensemble classifier is used to combine the predictions of several base classifiers built with a given learning algorithm in order to improve robustness over a single estimator.
It performs better than Logistic Regression and performance and performance is similar to SVM and RF over MNIST test dataset.



```
----------------ENSEMBLE CLASSIFIER-----------------

MNIST Accuracy = 94.87

USPS Accuracy = 34.24171208560428
```

Figure 15: Ensemble Classifier

8 Conclusion

In this project, An ensemble of four classifiers for a given task is implemented and the results of the individual classifiers are combined to make a final decision.

References

Pattern Recognition and Machine Learning by Christopher M. Bishop

https://scikit-learn.org/stable/auto$_examples/model_selectionhttps : //scikit - learn.org/stable/modules/ensemble.html$